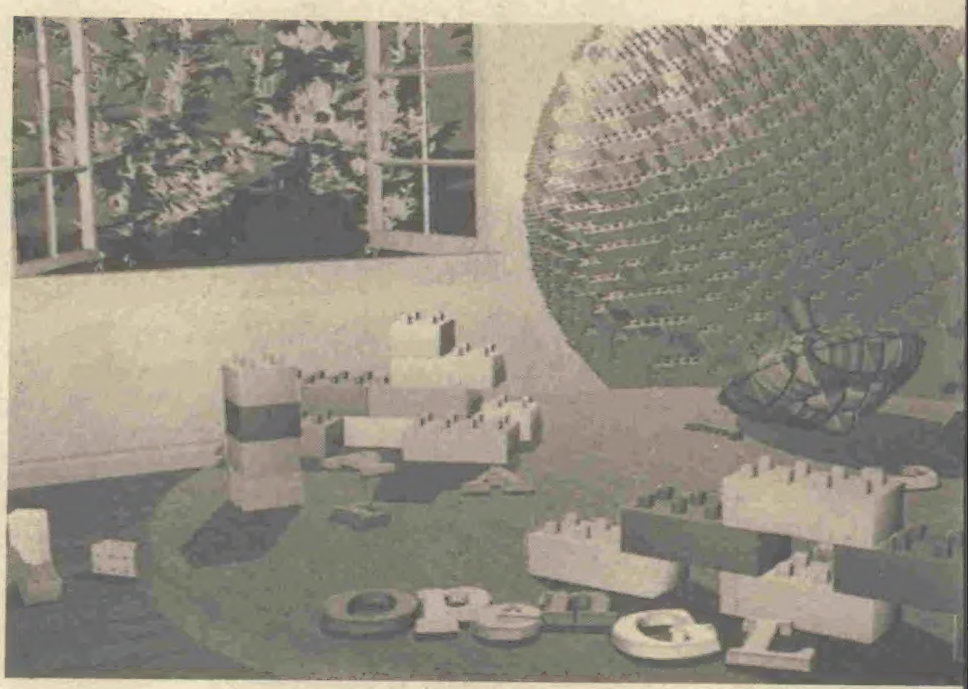
บุญเลิศ เอี่ยมทัศนา

วาดภาพเก็บไว้ใน เมตาไฟล์

ห่างหายไปนานกับเรื่องราวของเมตาไฟล์ ตอนนี้ขอเริ่มต้นใหม่และทบทวนของเก่า



เรื่องของเมตาไฟล์มีผู้นำมากล่าวถึงน้อยมากเมื่อเทียบกับบิตแม็พ และสำหรับผมโปรแกรมที่จะใช้วาดภาพเก็บไว้ในเมตาไฟล์ยิ่งหายากขึ้นไปอีก หากคุณประสบปัญหาเช่นเดียวกับผม บทความนี้อาจช่วยได้

เมตาไฟล์

ผมได้เคยนำเรื่องของเมตาไฟล์มากล่าวถึงแล้ว แต่ยังไม่สมบูรณ์ต้องพักไปเขียนโปรแกรมการวาดภาพก่อน ดังที่นำมาใช้ในการวาดภาพเก็บไว้ในบิตแม็พ ตอนนี้จึงเป็นของเมตาไฟล์บ้าง โดยที่ทิ้งไปนานจึงขอเริ่มต้นใหม่เป็นการทบทวนของเก่าอย่างสั้น ๆ ครับ

ชนิดของเมตาไฟล์

เมตาไฟล์แบ่งออกเป็น 2 ชนิดคือ เมตาไฟล์ธรรมดา และเพลซเอเบิลเมตาไฟล์ ซึ่งเมื่อโหลดลงหน่วยความจำแล้วจะอยู่ในฐานะเมตาไฟล์เช่นกัน ข้อแตกต่างที่สำคัญอันหนึ่งจึงอยู่ตรงการโหลด

เมตาไฟล์ธรรมดาแยกได้เป็นเมตาไฟล์ในดิสค์ และเมตาไฟล์ในหน่วยความจำ ข้อแตกต่างในการใช้งานคือ

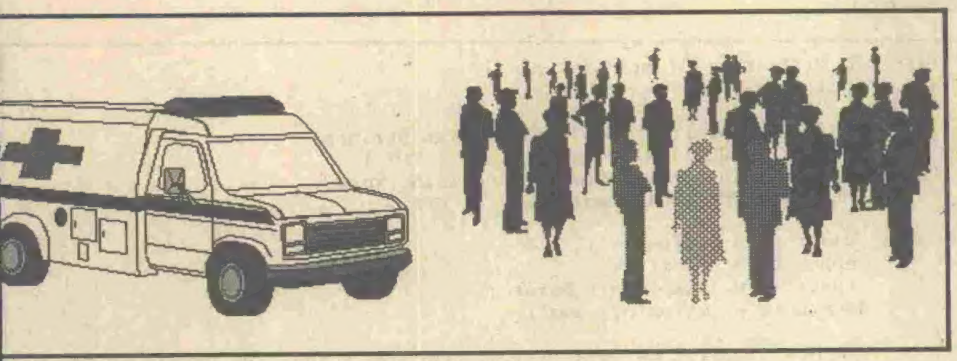
- เดิมที่หน่วยความจำน้อยอาจใช้แต่เมตาไฟล์ในดิสค์
- เมื่อให้ผู้ใช้วาดภาพ อาจเก็บภาพที่วาดไว้ในเมตาไฟล์ในหน่วยความจำเมื่อเสร็จงานแล้วจึงเก็บลงเมตาไฟล์ในดิสค์
- ในปัจจุบันที่ใช้กันมากคือ การส่งผ่านเมตาไฟล์ในหน่วยความจำทางคลิปบอร์ด และเมตาไฟล์ในดิสค์ทาง OLE (แต่ผมยังไม่เห็นโปรแกรมนี้)

ในด้านการใช้งานสามารถแบ่งได้เป็น เมตาไฟล์ภาคบันทึกเพื่อการบันทึกภาพ กับเมตาไฟล์ภาคแสดงเพื่อการนำภาพที่บันทึกไว้ออกแสดง

ลักษณะของภาพ

ภาพที่บันทึกไว้ในเมตาไฟล์เท่าที่ผมพบมีลักษณะเป็นภาพลายเส้นดังรูปที่ 1 และคำว่าลายเส้นผมขอใช้ในสองความหมาย คือเป็นภาพลายเส้น กับเป็นการวาดภาพด้วยเครื่องมือที่เป็นดินสอ เป็นการวาดภาพโดยเลื่อนเม้าส์ไปเหมือนการวาดด้วยดินสอธรรมดา เช่น วาดเป็นรูปคิ้วของรถยนต์

และอีกกรณีหนึ่งซึ่งขอเรียกเป็นลายเส้นปิด ดังรูปคนจะเดินเส้นเป็นขอบเช่น ด้วยสีน้ำเงินจนมาบรรจบกันแล้วให้ระบายสีภายในด้วยสีน้ำเงินเช่นกัน หรืออย่างรูปกาชาดจะเดินขอบด้วยสีดำ ระบายสีภายในด้วยสีแดง



ปที่ 1 ลักษณะของภาพในเมตาไฟล์

ขั้นตอนการใช้เมตาไฟล์

การใช้งานเมตาไฟล์มีขั้นตอนตามที่ผมนำส่วนของโปรแกรมในเรื่องนี้มาแสดงในลิสติงที่ 1 กล่าวคือ

การสร้างเมตาไฟล์

อยู่ในโปรแกรมของกิจกรรม wmRButtonDown คือ

1. ให้กำเนิดเมตาไฟล์ในดิสค์ด้วยฟังก์ชัน CreateMetaFile (หากต้องการให้เป็นเมตาไฟล์ในหน่วยความจำ ให้ผ่านค่า nil แทนชื่อไฟล์) จะได้เมตาไฟล์ภาคบันทึกใน recDC

2. บันทึกภาพที่ต้องการ ตามตัวอย่างนี้ให้บันทึกภาพวงรี 1 วง กับข้อความ 1 บรรทัด เพื่อแสดงว่าบันทึกได้ทั้งคู่ จะเห็นว่าเหมือนกับการให้เอาต์พุตทางดีซี แต่เป็นการบันทึก

3. ปิดเมตาไฟล์ภาคบันทึก เมื่อบันทึกเสร็จแล้วให้ปิดเมตาไฟล์ด้วย CloseMetaFile จะเป็นการเขียนลงดิสค์จริง และได้ค่าเมตาไฟล์ภาคแสดงใน hPlay

4. นำออกแสดง หากต้องการนำออกแสดง เช่น ทางจอภาพกับดีซีของหน้าต่าง ให้แสดงด้วยฟังก์ชัน PlayMetaFile

5. ยกเลิกเมตาไฟล์ภาคแสดง เมื่อไม่ต้องการใช้งานเมตาไฟล์ภาคแสดงแล้ว ให้ยกเลิกเมตาไฟล์ภาคแสดงด้วยฟังก์ชัน DeleteMetaFile สำหรับเมตาไฟล์ภาคบันทึกจะถูกยกเลิกโดยปริยายเมื่อให้ปิดเมตาไฟล์ นำมาใช้บันทึกอีกไม่ได้

การอ่านเมตาไฟล์

เมื่อมีเมตาไฟล์ในดิสค์แล้ว สามารถอ่านมาแสดงได้ตามโปรแกรมของกิจกรรม wmLButtonDown คือ

1. อ่านเมตาไฟล์ในดิสค์ด้วยฟังก์ชัน GetMetaFile จะได้เป็นเมตาไฟล์ภาคแสดงใน hPlay

2. ที่ให้ตรวจว่ามีไฟล์นี้หรือไม่ เพราะฟังก์ชัน GetMetaFile ไม่ตรวจ และถ้าไม่มีไฟล์ที่ให้อ่านจะเกิด runtime error

โปรแกรม MtfPaint

จากความรู้ดังกล่าวผมนำมาเขียนโปรแกรม MtfPaint เพื่อการวาดภาพเก็บไว้ในเมตาไฟล์ธรรมดา ดังโปรแกรมในลิสติงที่ 2 เมื่อรันโปรแกรมนี้จะแสดงหน้าต่างดังรูปที่ 2

ท้าวความ

จะเห็นว่าภาพหน้าต่างเหมือนของโปรแกรม BmpPaint มาก เพราะใช้ซึ่งผมคงต้องขอเรียกว่ากราฟิกเอดิเตอร์เช่นกัน คือตามที่ผมได้ทำเป็นสกุล ToolWndC กับ PalWndC ไว้ในยูนิต UCellWn1

แล้วนำมาใช้เป็นกล่องเครื่องมือเป็นจานสีในสกุล CanvasWndC ใน

ลิสติงที่ 1

```
CONST  MtfFNK = 'c:\simmtf.mtf';

PROCEDURE  WndProC.wmRButtonDown(var msg : TMessage);
VAR  recDC,wndDC : HDC;  hPlay : THandle;
BEGIN
  recDC := CreateMetaFile(MtfFNK);
  Ellipse(recDC, 20,20,200,150);
  TextOut(recDC, 50, 160, 'SIMPLE METAFILE', 15);
  hPlay := CloseMetaFile(recDC);
  wndDC := GetDC(HWindow);  PlayMetaFile(wndDC, hPlay);
  ReleaseDC(HWIndow, wndDC);  DeleteMetaFile(hPlay);
END;


PROCEDURE  WndProC.wmLButtonDown(var msg : TMessage);
VAR  fiH : Integer;  hPlay : THandle;  wndDC : HDC;
BEGIN
  fiH := _lopen(MtfFNK, of_Read);
  if  fiH <= 0  then  begin  MessageBeep(100);  Exit  end
  else  _lclose(fiH);
  hPlay := GetMetaFile(MtfFNK);
  wndDC := GetDC(HWindow);  PlayMetaFile(wndDC, hPlay);
  ReleaseDC(HWIndow, wndDC);  DeleteMetaFile(hPlay);
END;
```

ยูนิต UCanvas1 ซึ่งเป็นสกุลหน้าต่างวาดภาพ เมื่อผู้ใช้วาดภาพจะนำออกแสดงและเก็บไว้ใน recDC ซึ่งเป็นดีซี โดยที่

- ในโปรแกรม BmpPaint เป็นเมมดีซีที่มีผืนภาพบิตแม็พ (เปล่าหรือมีภาพอยู่แล้ว)

- ส่วนของโปรแกรม MtfPaint จะเป็นเมตาไฟล์ภาคบันทึก

การวาดภาพ

เพื่อให้ผู้ใช้วาดภาพได้ตามที่กล่าวมาในโปรแกรมนี้ได้ให้ดำเนินการดังนี้

1. ได้ให้ WndProC ของโปรแกรมนี้สืบสกุลจากสกุล CanvasWndC (เช่นเดียวกับของโปรแกรม BmpPaint)

2. ใน SetupWindow ได้ให้กิจกรรม SetNewMTF เตรียมเมตาไฟล์ภาคบันทึกให้แก่ recDC ใน CanvasWndC

เตรียมเมตาไฟล์ภาคบันทึก

การเตรียมเมตาไฟล์ภาคบันทึกอยู่ในโปรแกรมของกิจกรรม SetNewMTF โดยที่

1. ได้ให้ผ่านค่ามาดังนี้

- mde เป็นแม็พปิงโหมด ซึ่งของเมตาไฟล์ธรรมดาควรอย่างยิ่งที่จะให้เป็น mm_Text และเมื่อเป็นโหมดนี้ให้ผ่านค่าสเกลแก่ scle เป็นศูนย์เพราะไม่ใช้

- ค่า px1 ถึง py2 เป็นค่าจุดมุมของพื้นที่ภาพ และถ้าต้องการกำหนดเขตพื้นที่ (regin) เพื่อไม่ให้ผู้ใช้วาดภาพออกไปนอกพื้นที่ภาพจะต้องให้ค่า isrgn เป็นจริง

2. ให้กำเนิดเมตาไฟล์ในหน่วยความจำ เป็นเมตาไฟล์ภาคบันทึกให้แก่ recDC ใน CanvasWndC โดยตรงเป็นเมตาไฟล์ว่าง

3. เรียกใช้ PrepScale ของ Can-

ลิสติงที่ 2

```
PROGRAM  MtfPainT;    { mtfpaint.pas }
{$R mtfpaint.res}

{USES    WinTypes,WinProcs,WObjects,StdDlgs,Strings,
         UBkUtils,UCellWn1,UCanvas1; }    { TPW }
USES     Objects,WinTypes,WinProcs,OWindows,Strings,
         UBkUtils,UCellWn1,UCanvas1;      { BPW }

CONST   classnme = 'mtfpaint';
        appname  = 'simmtf';
        thetitle = 'Simple MTF Paint';
        menuname = 'mtfpaint_menu';

        cm_New  = 101;  cm_Open  = 102;  cm_Save = 103;  cm_SaveAs = 104;
        cm_Quit = 109;  cm_PenSi = 206;


TYPE
  WndProCP = ^WndProC;
  WndProC = OBJECT(CanvasWndC)
    mtfFNB : array[0..79] of Char;
    hPlay  : THandle;
    twopnt : Integer;
 constructor  Init(apar : PWindowsObject; atitle : PChar);
 procedure  Destroy; virtual;
 function   GetClassName: PChar; virtual;
 procedure  SetupWindow;  virtual;
 procedure  DoPaint(paintDC : HDC); virtual;
 procedure  wmCommand(var msg : TMessage);
             virtual wm_First + wm_Command;
 procedure  LoadSimMtf(finme : PChar);
 procedure  DoSaveAs;
 procedure  SaveSimMtf(finme : PChar);

 procedure  SetNewMTF(mde,scle,px1,py1,px2,py2 : Integer; isrgn : Boolean);
 procedure  CloseFileForPlay;
            END;

(* ================= WndProC procedure ============== *)
const  notbmp = false;

CONSTRUCTOR  WndProC.Init(apar : PWindowsObject; atitle : PChar);
BEGIN
  CanvasWndC.Init(apar, atitle, tray16hor, notbmp);
  attr.menu := LoadMenu(HInstance, menuname);
  attr.Style := Attr.Style or ws_VScroll or ws_HScroll;
  Scroller := New(PScroller, Init(@Self, 10, 10, 0, 0));
  recDC := 0;  hPlay := 0;
  lstrcpy(mtfFNB, 'd:\book2\14mtf\wmffi\*.wmf');
END;


const  havergn = true;
PROCEDURE  WndProC.SetupWindow;
VAR    aDC : HDC;  curnum : Integer;
       itool : ToolTpeT;  temcur : HCursor;
BEGIN
  CanvasWndC.SetupWindow;
  SetNewMTF(mm_Text, 0, 100, 50, 400, 250, havergn);
END;


PROCEDURE  WndProC.Destroy;
BEGIN
  if  recDC <> 0  then  DeleteMetaFile(CloseMetaFile(recDC));
  if  hPlay <> 0  then  DeleteMetaFile(hPlay);
  CanvasWndC.Destroy;
END;


FUNCTION  WndProC.GetClassName : PChar;
BEGIN  GetClassName := classnme   END;
```

vasWndC เพื่อให้ค่าต่าง ๆ

4. ในการนี้ได้ให้ยุบเมตาไฟล์ภาคบันทึกคือ recDC ของเดิมหากมีข้อมูล และเมตาไฟล์ภาคแสดงคือ hPlay หากมีข้อมูลเช่นกัน

เท่านี้ก็พร้อมที่จะให้ผู้ใช้วาดภาพแล้ว

อ่านจากไฟล์

แต่ถ้าผู้ใช้ต้องการอ่านจากไฟล์เพื่อดูหรือวาดภาพต่อ ได้จัดเป็นรายการ File/Open ไว้ มาทำงานที่โปรแกรมของกิจกรรม LoadSimMtf ซึ่งจะ

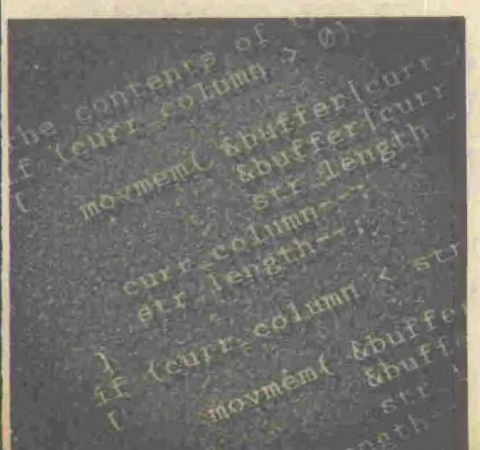
1. แสดงไฟล์ไดอะล็อกบ็อกซ์ให้ผู้ใช้เลือกไฟล์ นำมาตรวจสอบว่ามีไฟล์นี้หรือไม่ เพราะผู้ใช้อาจป้อนชื่อไฟล์

2. หากมีไฟล์ให้อ่านด้วย GetMetaFile ได้เป็นเมตาไฟล์ภาคแสดงให้แก่ hLoad

3. ให้กิจกรรม SetNetMTF ให้กำเนิดเมตาไฟล์ภาคบันทึกเป็นไฟล์เปล่าให้แก่ recDC และเซตค่าต่าง ๆ ดังที่กล่าวมา

4. ถ่ายทอดข้อมูลจากเมตาไฟล์ภาคแสดงให้แก่เมตาไฟล์ภาคบันทึกด้วยวิธีเดียวกับที่นำออกแสดงในหน้าต่างคือที่ใช้ฟังก์ชัน PlayMetaFile

5. ยุบเมตาไฟล์ภาคแสดงคือ hLoad เพราะได้ถ่ายทอดข้อมูลให้แก่ recDC แล้ว ไม่ต้องการใช้อีก เมตาไฟล์ทั้งสองในขณะนี้จะต้องถือว่าเป็นคนละไฟล์กัน

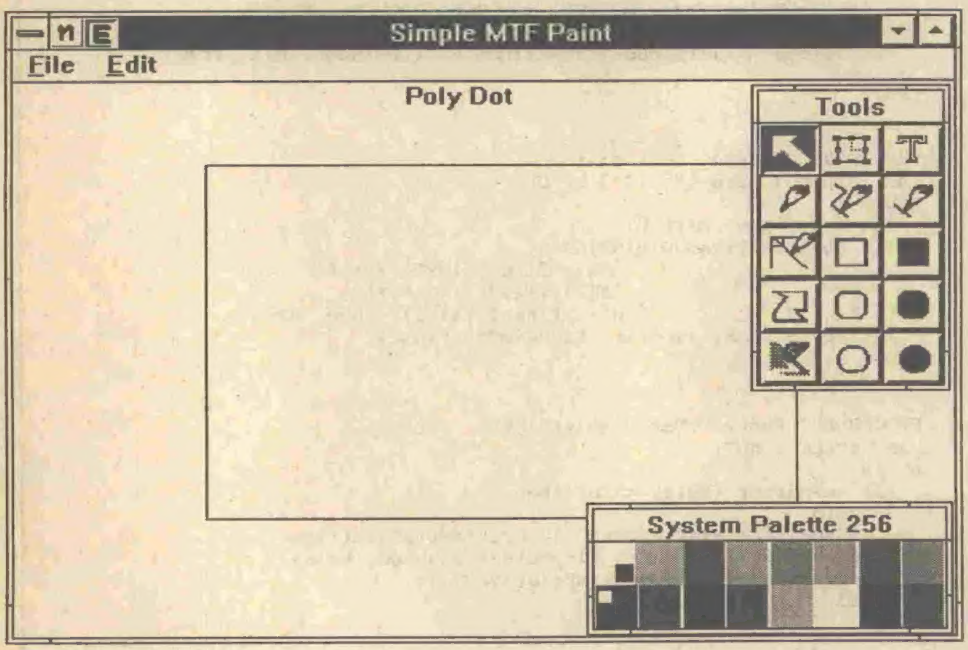


ลิสติงที่ 2 (ต่อ)

```
PROCEDURE  WndProC.DoPaint(paintDC : HDC);
BEGIN
  Rectangle(paintDC, pctxy1.x-twopnt, pctxy1.y-twopnt,
                     pctxy2.x+twopnt, pctxy2.y+twopnt);
  if  hasrgn  then  SelectClipRgn(paintDC, wndRgn);
  if  newpat  then  CloseFileForPlay;
  if  hPlay <> 0  then  PlayMetaFile(paintDC, hPlay);
END;


PROCEDURE  WndProC.wmCommand(var msg : TMessage);
BEGIN
  if  msg.lParamLo <> 0  then  Exit;
  case  msg.wParam  of
   cm_New    :
     begin  SetNewMTF(mm_Text, 0, 100, 50, 400, 250, havergn);
            InvalidateRect(HWindow, nil, true)  end;
   cm_Open   : LoadSimMtf(mtfFNB);
   cm_Save   :
     if  (StrScan(mtfFNB, '*') = nil) and
         (StrScan(mtfFNB, '?') = nil) then  SaveSimMtf(mtfFNB)
     else  DoSaveAs;
   cm_SaveAs : DoSaveAs;
   cm_Quit   : CloseWindow;
  else
   CanvasWndC.wmCommand(msg);
  end;
END;


PROCEDURE  WndProC.LoadSimMtf(finme : PChar);
VAR  fiH : Integer;  adc : HDC;  hLoad : THandle;  mde,scle : Integer;
     tem1 : LongInt;  buf : array[0..127] of Char;
BEGIN
  lstrcpy(buf, finme);
  if  not  GetFiNameDlg(HWindow,
                        'Meta file (*.WMF)¦*.wmf¦' +
                        'All files (*.*)¦*.*¦',
                        nil, buf, true)  then  Exit;
  fiH := _lopen(buf, of_Read);
  if  fiH > 0  then  _lclose(fiH)  else  Exit;

  lstrcpy(finme, buf);  hLoad := GetMetaFile(finme);
  if  hLoad <> 0  then
    begin
    SetNewMTF(mm_Text, 0, 100, 50, 400, 250, havergn);
    PlayMetaFile(recDC, hLoad);  newpat := true;
    DeleteMetaFile(hLoad); InvalidateRect(HWindow, nil, true);
    end
END;


PROCEDURE  WndProC.DoSaveAs;
VAR  finme : array[0..127] of Char;
BEGIN
  lstrcpy(finme, mtfFNB);
  if  not  GetFiNameDlg(HWindow,
                        'Meta file (*.WMF)¦*.wmf¦' +
                        'All files (*.*)¦*.*¦',
                        nil, finme, false)  then  Exit;
  lstrcpy(mtfFNB, finme);  SaveSimMtf(finme);
END;


PROCEDURE  WndProC.SaveSimMtf(finme : PChar);
VAR  sveDC : HDC;
BEGIN
  if  newpat or (hPlay <> 0) then
    begin
    CloseFileForPlay;  sveDC := CreateMetaFile(finme);
    if  hPlay <> 0  then  PlayMetaFile(sveDC, hPlay);
    DeleteMetaFile(CloseMetaFile(sveDC));
    end;
END;
```

ลิสติงที่ 2 (ต่อ)

```
PROCEDURE  WndProC.SetNewMTF(mde,scle,px1,py1,px2,py2 : Integer;
                                                  isrgn : Boolean);
VAR   arct : TRect;
BEGIN
  if  recDC <> 0  then  DeleteMetaFile(CloseMetaFile(recDC));
  if  hPlay <> 0  then  DeleteMetaFile(hPlay);  hPlay := 0;
  recDC := CreateMetaFile(nil);  newpat := false;
  SetRect(arct, px1, py1, px2, py2);  twopnt := 2;
  PrepScale(mde, scle, arct, isrgn);
END;


PROCEDURE  WndProC.CloseFileForPlay;
BEGIN
  if  hPlay <> 0  then  DeleteMetaFile(hPlay);
  hPlay := CloseMetaFile(recDC);
  recDC := CreateMetaFile(nil);
  if  hPlay <> 0  then PlayMetaFile(recDC, hPlay);
  newpat := false;
END;


(* ------------------  TheAppC Class  ---------------- *)

TYPE
  TheAppC = OBJECT(TApplication)
    procedure  InitMainWindow; virtual;
            END;


PROCEDURE  TheAppC.InitMainWindow;
BEGIN
  MainWindow := New(WndProCP, Init(nil,thetitle));
END;

(* --------------------  MAIN PROGRAM  ----------------- *)

VAR   app :  TheAppC;

BEGIN
  app.Init(appname);
  app.Run;
  app.Done;
END.

WVSPrintF(buf,'Error number %d',res);
```



รูปที่ 2 หน้าต่างของโปรแกรม MtfPaint มีกล่องเครื่องมือทางด้านขวาและจานสีด้านล่าง

การแสดงภาพ

ตามที่กล่าวแล้วว่าเมื่อผู้ใช้วาดภาพ เช่น วาดเป็นรูปวงกลม โปรแกรมใน CanvasWndC จะนำออกแสดงในหน้าต่างโดยตรงคือ ไม่ผ่าน Paint แต่เมื่อต้องเพนต์ เช่น เมื่อผู้ใช้เปลี่ยนขนาดของหน้าต่าง เราจะต้องแสดงภาพในหน้าต่างใหม่ จากภาพที่เก็บไว้ใน recDC ซึ่งถ้าเป็นบิตแม็พสามารถลอกภาพมาแสดงได้โดยตรง

สำหรับเมตาไฟล์จะต้องดำเนินการเป็นพิเศษ เริ่มจากกิจกรรม DoPaint เรียกใช้กิจกรรม CloseFileForPlay เพื่อที่จะ

1. ปิดเมตาไฟล์ภาคบันทึก เช่นเดียวกับที่เก็บลงในไฟล์ในดิสค์ในลิสติงที่ 1 แต่สำหรับกรณีนี้เพื่อให้ได้เมตาไฟล์ภาคแสดงให้แก่ hPlay
2. ให้กำเนิดเมตาไฟล์ภาคบันทึกใหม่แล้วถ่ายทอดข้อมูลจากภาคแสดงให้เพื่อเตรียมไว้สำหรับการบันทึกต่อ
3. เมื่อกลับไป DoPaint จะนำเมตาไฟล์ภาคแสดงออกแสดงกับเพนต์ดีซีปรากฏภาพในหน้าต่าง

ตรงนี้ครับที่ผมไม่ค่อยสบายใจนัก เพราะดูยอกย้อน ไม่ทราบว่าไมโครซอฟต์ต้องการให้เขียนโปรแกรมเช่นนี้หรือไม่ หาตัวอย่างไม่ได้ ที่มีกล่าวในหนังสือต่าง ๆ บ้างจะเป็นการบันทึกเพียงครั้งเดียวแสดงเพียงครั้งเดียว หากไม่ดำเนินการเช่นนี้ก็มองไม่เห็นทางอื่น

เก็บลงไฟล์

เมื่อผู้ใช้ต้องการเก็บลงไฟล์ หากต้องการชื่อไฟล์จะแสดงไดอะล็อกบ็อกซ์ให้ผู้ใช้ป้อนหรือเลือกชื่อไฟล์ แล้วไปเก็บลงไฟล์ใน SaveSimMtf ซึ่งจะ

1. ให้ CloseFileForPlay ปิดเมตาไฟล์ภาคบันทึกเป็นภาคแสดง เช่นเดียวกับการนำออกแสดง
2. ให้กำเนิดเมตาไฟล์ภาคบันทึก

ใหม่เป็นเมตาไฟล์ในดิสค์ตามชื่อไฟล์ที่ผ่านมาให้

3. ถ่ายทอดข้อมูลจากเมตาไฟล์ภาคแสดงให้แก่เมตาไฟล์ภาคบันทึก แล้วปิดเมตาไฟล์ภาคบันทึกพร้อมกับยุบเมตาไฟล์ภาคแสดงที่ได้มาจากการปิดไฟล์

รีซอร์ส

โปรแกรมนี้ความจริงผมเขียนเป็นโปรแกรมเพื่อประกอบคำอธิบายเท่านั้น สามารถนำไปใช้งานของเราเองได้ แต่จะใช้กับแอพพลิเคชันโดยทั่วไปไม่ได้ จะต้องเก็บไว้ในเพลชเอเบิลเมตาไฟล์ตามที่ผมจะนำมากล่าวถึงต่อไป

รายการที่เมนูจึงมีแต่ File กับ Edit แต่ Edit ยังไม่ได้ใช้ สำหรับรายการไฟล์เป็นรายการไฟล์มาตรฐาน ดังที่ได้กำหนดค่าคงตัวไว้เป็นค่าซีเอ็มตอนต้นของโปรแกรม ขอให้สร้างรีซอร์สตามนี้ด้วยครับ

ปรับปรุงโปรแกรมเดิม

มีเรื่องที่ค้างอยู่และผมก็หลงลืมด้วย คือ ในการวาดรูปที่เป็นลายเส้นปิด เช่น รูปคนตามรูปที่ 1 หากเป็นภาพบิตแม็พอาจให้ผู้ใช้วาดเส้นขอบก่อนแล้วระบายสีภายในด้วยการระบายสีพื้นด้วยเครื่องมือที่ทำเป็นรูปลูกกลิ้ง (อยู่ในโปรแกรม BmpPaint)

แต่ของเมตาไฟล์จะใช้วิธีนี้ไม่ได้ เพราะในเมตาไฟล์จะต้องจัดโปรแกรมให้เลื่อนภาพเป็นแพตเทอร์นได้ แพตเทอร์นหนึ่งคือคำสั่งที่ใช้วาดภาพ 1 คำสั่ง เช่น วงกลม 1 วง เป็นต้น แต่ตามที่กล่าวมาและตามโปรแกรมเดิมแยกเป็น 2 คำสั่ง คือการวาดเส้นขอบใช้ Polygon กับการระบายสีภายในใช้ ExtFloodFill

จะต้องทำให้เหลือเพียงคำสั่งเดียว คือใช้ Polygon นั่นคือจะต้องกำหนดเครื่องมือขึ้นใหม่ ซึ่งผมไม่ทราบว่าจะใช้ชื่อใดดีจึงให้ชื่อว่า PolyDot และเนื่องจากไม่ใช้ลูกกลิ้งทาสีจึงจะใช้ปุ่มนี้อยู่ตรงมุมล่างซ้ายของกล่องเครื่องมือตามรูปที่ 2 ได้เปลี่ยนเป็นปุ่มใหม่แล้ว ขอให้ดำเนินการดังนี้ครับ

รีซอร์ส

ขอให้สร้างรีซอร์สของบิตแม็พเป็นภาพที่แสดงความหมายว่าเป็นเครื่องมือวาดภาพลายเส้นปิด ให้ชื่อรีซอร์สเป็นหมายเลขหรือถ้อยคำ ผมให้เป็น POLYDOT เพื่อให้แตกต่างจากของเดิม เสร็จแล้วเก็บไว้ในไฟล์ tool.res (ของเดิม)

ในยูนิต UCellWn1

ให้เปลี่ยนแปลงและเพิ่มโปรแกรมที่เกี่ยวกับสกุล ToolWndC ดังนี้

1. ในการกำหนดแบบข้อมูล ToolTpeT และในการกำหนดค่าคงตัว ToolConv ให้เพิ่มคำว่า polydot และในการกำหนดค่าคงตัว ToolText ให้เพิ่มสตริง "PolyDot" ดังรูปที่ 3 ท่อนบน

```
TYPE  ToolTpeT = (arrw,txt,scis,pencil,polyline,lne,tsqre,
                  rctngl,frctngl,rndrct,frndrct,ellpe,fellpe,
                  poly,filler,polydot,mvepict,mvepat);
CONST ToolConv : array[0..15] of ToolTpeT =
       (arrw,scis,txt,pencil,polyline,lne,tsqre,rctngl,frctngl,
        poly,rndrct,frndrct,filler,ellpe,fellpe,polydot);

      ToolText : array[ToolTpeT] of PChar =
        ('Arrow','Text','Scissor','Pencil','Poly line','Line',
         'TSquare','Rectangle','Fill Rect','Round Rect',
         'Fill Round Rect','Ellipse','Fill Ellipse',
         'Polygon','Filler','Poly Dot',
         'Move picture','Move pattern');

{------------------------------------------------------------------------}

CONSTRUCTOR  ToolWndC.Init(apar : PWindowsObject; IDwin,xx1,yy1 : Integer;
                                                  isBmp : Boolean);
VAR  clldta : CellDtaT;  itool : Integer;  temBmp : HBitmap;
BEGIN
  CellWndC.Init(apar, 'Tools', IDwin);
  with  clldta  do
    begin
    attrx := xx1;  attry := yy1;  attrw := 90; attrh := 150;
    logwid := 600;  loghi := 1000;  xxcell := 3; yycell := 5;
    lftdef := 0;  rgtdef := -1;
    end;
  SetCell(clldta);  isBmpF := isBmp;
  for  itool := fsttool to lsttool  do
    toolary[itool] := LoadBitmap(HInstance, PChar(itool+101));
  if  not isBmpF  then
    begin   DeleteObject(toolary[12]);
            toolary[12] := LoadBitmap(HInstance, 'POLYDOT')  end;
END;

{------------------------------------------------------------------------}

PROCEDURE  ToolWndC.NewSelect(butt,cellnum : Integer);
VAR  tem : TemT;
BEGIN
  if  not isBmpF  then  if  cellnum = 12  then  cellnum := 15;
  tem.llo := Ord(ToolConv[cellnum]);  tem.hhi := butt;
  if  (butt = lftbutt) or (butt = rgtbutt)  then
    SendMessage(hToWnd, wm_UCellWndSel, wndID, LongInt(tem))
END;


PROCEDURE  ToolWndC.CursorPstn(cellnum : Integer);
VAR  tind : ToolTpeT;  buff : array[0..79] of Char;
BEGIN
  if  not isBmpF  then  if  cellnum = 12  then  cellnum := 15;
  tind := ToolConv[cellnum];
  lstrcpy(buff, ToolText[tind]);
  lstrcat(buff, '  ');
  SendMessage(hToWnd, wm_UCellWndInfo, wndID, LongInt(@buff));
END;
```

รูปที่ 3 ข้อความที่ใช้แก้โปรแกรมในยูนิต UCellWn1

2. ในการกำหนดสกุล ToolWndC ให้เพิ่มฟิลด์ข้อมูล isBmpF เป็นบูลีน และให้เพิ่มค่าพารามิเตอร์ของ Init เป็น isBmp เป็น Boolean เช่นกัน

3. ในโปรแกรมของกิจกรรม ToolWndC.Init ให้เพิ่มโปรแกรมเป็นดังรูปที่ 3 ท่อนกลางคือ

- ให้ค่า isBmp แก่ isBmpF
- โหลดบิตแม็พรีซอร์ส POLYDOT มาแทนของเดิมในอะเรย์ toolary ช่องที่ 12

4. ในโปรแกรมของกิจกรรม NewSelect และ CursorPstn ให้เพิ่มโปรแกรมตามรูปที่ 3 ท่อนล่าง บรรทัดแรกของแต่ละกิจกรรม ความหมายคือ ถ้าเป็นเมตาไฟล์ หากผู้ใช้เลือกปุ่มที่ 12 ให้ถือว่าเลือกปุ่มที่ 15

ในยูนิต UCanvas1

คือในสกุล CanvasWndC ขอให้เพิ่มโปรแกรมดังนี้

1. เพิ่มฟิลด์ข้อมูล newtool ต่อจาก tool มีแบบข้อมูลเป็น ToolTpeT และใน Init ให้ค่าเป็น arrw

2. ในกิจกรรม wmUCellWndSel ให้เพิ่มโปรแกรมดังรูปที่ 4 ท่อนแรก ความหมายคือ เมื่อเครื่องมือ (tool) เป็น polydot ให้เปลี่ยนค่าเป็น pencil คือดินสอ เพราะมีวิธีวาดภาพในตอนต้นนี้เหมือนกัน

3. ในกิจกรรม DoPencil ให้เพิ่มโปรแกรมตามรูปที่ 4 ท่อนล่าง เพื่อให้ตรวจดูว่าถ้าผู้ใช้เลือกเครื่องมือเป็นลายเส้นปิด จะให้วาดภาพด้วย Polygon ทั้งในหน้าต่าง ซึ่งจะเป็นการวาดทับของเดิมเพื่อระบายสีภายใน และเก็บไว้ในเมตาไฟล์ แต่ถ้าไม่ใช่คือผู้ใช้เลือกเครื่องมือเป็นดินสอ คงให้เก็บไว้ในเมตาไฟล์อย่างเดียว (ตามเดิม)

เท่านี้ครับจะได้เครื่องมือใหม่ไว้ใช้ ขอให้ลองวาดดู จะเป็นการเดินเส้นด้วยปากกาและระบายสีภายในด้วยสีของแปรง ซึ่งในการเดินเส้นไม่จำเป็นต้องนำมาบรรจบกัน เพราะ Polygon จะต่อเส้นสุดท้ายให้

```
PROCEDURE  CanvasWndC.wmUCellWndSel(var msg : TMessage);
BEGIN
 case  msg.wParam  of
  id_PenSel    : penco := msg.lParam;
  id_BrushSel : fillco := msg.lParam;
  id_Tool      :
    begin
    if  (tool = poly) or (tool = polyline) then  DrawPoly(false);
    tool := ToolTpeT(msg.lParamLo);
    if  tool = polydot  then
      begin  tool := pencil;  newtool := polydot  end
    else  newtool := arrw;
    curcur := curary[tool]; { frstkey := true;}
    end;
 end;  { case }
END;

-------------------------------------------------------------------

PROCEDURE  CanvasWndC.DoPencil;
VAR  abrsh,oldbrsh,orecbrsh : HBrush;
BEGIN
  if  newtool = polydot  then
    begin
    abrsh := CreateSolidBrush(fillco);
    oldbrsh := SelectObject(drawDC, abrsh);
    orecbrsh := SelectObject(recDC, abrsh);
    Polygon(drawDC, pntcltn^.CollectPnts^, pntcltn^.NumPoint);
    Polygon(recDC, pntcltn^.CollectPnts^, pntcltn^.NumPoint);
    SelectObject(recDC, orecbrsh);
    DeleteObject(SelectObject(drawDC, oldbrsh));
    end
  else  PolyLine(recDC, pntcltn^.CollectPnts^, pntcltn^.NumPoint);
  Dispose(pntcltn, Done);  pntcltn := nil;
  polyfrst := true;
END;
```

รูปที่ 4 ข้อความที่ใช้แก้โปรแกรมในยูนิต UCanvas1

โอโอพี

เมื่อผมเขียนโปรแกรมสร้างสกุลในกลุ่มสกุลกราฟิกเอดิเตอร์คือ CellWndC, ToolWndC, PalWndC และ CanvasWndC แล้วนำมาใช้ในโปรแกรม BmpPaint อาจมองไม่เห็นความสำคัญของโอโอพี เพราะเดินเรื่องติดต่อกันมาเป็นเนื้อเดียว

แต่เมื่อผมนำมาใช้ในโปรแกรม MtfPaint คงทำให้เห็นข้อดีของโอโอพีชัดเจนขึ้น เพราะผมไม่ต้องเขียนโปรแกรมในส่วนนี้อีก สามารถนำของเก่ามาใช้ได้ทันที (ที่ต้องแก้โปรแกรมเพราะทำตกหล่นไป)

และจากนี้หากต้องการเก็บภาพไว้ในสิ่งใดอีกที่เป็นดีซี สามารถนำมาใช้ได้ทันที เช่น ในวินโดวส์เอ็นทีมีเมตาไฟล์ชนิดใหม่เรียกว่า Enhanced Metafile ก็สามารถใช้โปรแกรมกราฟิกเอดิเตอร์นี้ได้ และความจริงคือจากโปรแกรม MtfPaint นี้เอง โดยเปลี่ยนคำที่ใช้โดยเพิ่มคำว่า Enh เช่น CreateMetaFile เป็น CreateEnhMetaFile แต่ทั้งนี้คงต้องรอให้เทอร์โบปาสคาลมียูนิตที่มีคำในวินโดวส์เอ็นทีก่อนครับ

ในฉบับหน้า

หากคุณมีไฟล์ wmf ซึ่งถ้ามีก็จะมีเป็นจำนวนมาก ต้องการนำออกแสดงในวิธีเดียวกับเครื่องฉายภาพนิ่ง คือแสดงภาพติดต่อกันไป จะอยู่ในฉบับหน้า พร้อมด้วย drag and drop ส่วนการวาดภาพเก็บไว้ในเพลชเอเบิลเมตาไฟล์ต้องต่อไปอีก 1 ฉบับครับ